

日 本 国 特 許 庁
JAPAN PATENT OFFICE



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application: 2001年 3月14日

出 願 番 号

Application Number: 特願2001-071680

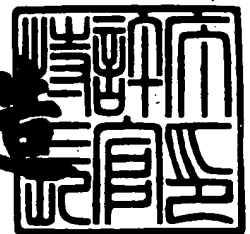
出 願 人

Applicant(s): 株式会社東芝

2001年 7月27日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



出証番号 出証特2001-3065013

【書類名】 特許願

【整理番号】 1FB00Y0051

【あて先】 特許庁長官殿

【国際特許分類】 G06F 13/00

【発明の名称】 クラスタシステム

【請求項の数】 4

【発明者】

 【住所又は居所】 東京都府中市東芝町 1 番地 株式会社東芝 府中事業所
内

 【氏名】 平山 秀昭

【特許出願人】

 【識別番号】 000003078

 【氏名又は名称】 株式会社 東芝

【代理人】

 【識別番号】 100083161

 【弁理士】

 【氏名又は名称】 外川 英明

 【電話番号】 (03)3457-2512

【手数料の表示】

 【予納台帳番号】 010261

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 クラスタシステム

【特許請求の範囲】

【請求項 1】

複数の電子計算機と、共有ディスク装置と、前記複数の電子計算機上で動作するプロセスから前記共有ディスク装置に記録されたファイルに対して共有アクセスするためにロック機能によりデータの一貫性を保持するための排他制御をするクラスタ共有ファイルシステムとを持ったクラスタシステムであって、

前記クラスタ共有ファイルシステムが管理するファイルを前記プロセスのアドレス空間内にマッピングして共有メモリを設定するためのクラスタ共有メモリ設定手段と、

前記クラスタ共有ファイルシステムのロック機能を前記共有メモリにアロケーションして前記共有メモリ上でデータの一貫性を保持するための排他制御を可能とするクラスタ共有メモリ用ロックアロケーション手段とを備えたことを特徴とするクラスタシステム。

【請求項 2】

前記設定された共有メモリ内の全てのページに対するアクセスを禁止にするアクセス禁止設定手段と

このアクセス禁止が設定されたページをアクセスしたときに R E A D ページフォールトが発生した場合、そのページに前記マッピングされたデータを前記共有ディスク装置に記録されたファイルから読み出して書き込むデータ書き込み手段と、

このデータが書き込まれたページを読み出し可能に設定する設定手段とを具備することを特徴とする請求項 1 記載のクラスタシステム。

【請求項 3】

前記設定された共有メモリ内の全てのページに対するアクセスを禁止にするアクセス禁止設定手段と

このアクセス禁止が設定されたページをアクセスしたときに W R I T E ページフォールトが発生した場合、そのページに前記マッピングされたデータを前記共

有ディスク装置に記録されたファイルから読み出して書き込むデータ書き込み手段と、

このデータが書き込まれたページを読み出し／書き込み可能に設定する設定手段とを具備することを特徴とする請求項 1 記載のクラスタシステム。

【請求項 4】

前記クラスタ共有メモリをロック操作した場合、前記クラスタ共有ファイルシステムが管理するファイルを前記プロセスのアドレス空間にマッピングしたクラスタ共有メモリ領域のページをアクセス不許可に設定してロックを取得する手段と、

この取得したロックをアンロック操作した場合、前記クラスタ共有メモリ領域の更新されたページのデータを前記クラスタ共有ファイルシステムが管理するファイルに書き戻す手段を具備したことを特徴とする請求項 3 記載のクラスタシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、記憶装置と計算機を専用の高速ネットワークで接続した SAN (Storage Area Network (ストレージエリア・ネットワーク)) 環境における並列処理プログラミングが容易なクラスタシステムに関するものである。

【0002】

【従来の技術】

近年、データのストレージ装置としての磁気ディスク装置と電子計算機（以下、サーバーコンピュータまたは単にサーバーと呼ぶ）とを Fiber Channel（ファイバ・チャネル）等の専用の高速ネットワークで接続した SAN が注目を浴びている。

【0003】

この SAN には以下のような利点がある。（１）複数のサーバーがストレージ装置を共有することができる。（２）ストレージ装置のアクセス負荷を LAN（

Local Area Network) から分離させることができる。(3) ファイバ・チャネル等によりストレージ装置へのアクセスを高速にすることができる。このうち(1)はSANの一般的な利点であるが、ここにクラスタ共有ファイルシステムの技術を用いると、単に複数のサーバーがストレージ装置を共有できるだけでなく、ファイルを共有アクセスすることが可能になる。

【0004】

米国Sistina Software Incが開発しているGFS(Global File System)は、このようなクラスタ共有ファイルシステムの一例である。クラスタ共有ファイルシステムでは、複数のサーバーコンピュータによるストレージ装置に記憶されたファイルへの共有アクセスを可能にしている。

【0005】

複数のサーバーコンピュータによるファイルへの共有アクセスというと、一般的にはNFS(Network File System)が連想されるが、NFSでは複数のサーバーコンピュータで同一ファイルを更新した場合のデータの一貫性を保証していない。これに対しクラスタ共有ファイルシステムではデータの一貫性を保証している。

【0006】

クラスタ共有ファイルシステムは、複数のサーバーコンピュータによるファイルの共有アクセス(READ/WRITE)機能の他に、データの一貫性を保証するために複数のサーバーコンピュータに跨るロック機構であるクラスタ共有ファイルシステム用ロック機構を持っている。このクラスタ共有ファイルシステム用ロック機構により、複数のサーバーコンピュータから構成されるクラスタシステム(Cluster System)上で並列アプリケーションプログラムが実行可能となる。

【0007】

例えば、クラスタ共有ファイルシステムとクラスタ共有ファイルシステム用ロック機構とを実装した複数のサーバーコンピュータがSANを介して1つの磁気ディスク装置に接続されているクラスタシステムについて考察する。このクラス

タシステムは、主メモリを共有していない疎結合のクラスタシステムである。各サーバーコンピュータ上で実行されているプロセスは、クラスタ共有ファイルシステムを用いることにより、磁気ディスク装置に記憶されているファイルに共有アクセスすることができる。しかも、プロセスは、クラスタ共有ファイルシステム用ロック機構を用いて排他制御処理をすることにより、磁気ディスク装置に記憶されているファイルのデータの一貫性を保証することができる。

【0008】

これに対して、ファイルシステムと共有メモリシステムとロック機構とが実装され、複数のプロセッサを持ち、例えば1つの磁気ディスク装置が接続されているSMP型（Symmetrical Multiprocessor）並列計算機について考察する。このSMP型並列計算機上で実行されている複数のプロセスは、それぞれファイルシステムを介して磁気ディスク装置に記憶されたファイルに共有アクセスしたり、共有メモリシステムを介して共有メモリ（主メモリ）にアクセスすることができる。また、複数のプロセスは、ロック機構を介して磁気ディスク装置に記憶されたファイルや共有メモリに記憶されたデータに対する共有アクセスにおける排他制御処理をすることにより、データの一貫性を保持することができる。

【0009】

このように従来の疎結合のクラスタシステムとSMP型並列計算機とを比較すると、両者ともに磁気ディスク装置に記憶されたファイルに対する共有アクセスおよびこれらに対するデータの一貫性の保持は可能である。しかし、クラスタシステムでは、複数のサーバーコンピュータから共有メモリへの共有アクセスをすることができない。

【0010】

【発明が解決しようとする課題】

クラスタ共有ファイルシステムを持つクラスタシステムでは、ファイルへの共有アクセスはできるが、メモリへの共有アクセスはできない。このため、クラスタ共有ファイルシステムを持つクラスタシステムでは、SMP型並列計算機と比べると、その実行するアプリケーションプログラムを並列プログラムにて記述す

るのが困難であるという課題がある。具体的には、複数のサーバーコンピュータ上で実行されているプロセスでファイル（又はデータ）を共有して互いにデータの一貫性を確保しながらデータの同期をとってデータ処理するプログラミングを記述する場合には、プロセスが共有して処理するファイル（又はデータ）をメモリ上に配置して処理することが必要であるが、これを行うことができない。もし、プロセスがファイル（又はデータ）を同期をとりながらデータの一貫性を確保して共有処理することが必要な場合には、クラスタ共有ファイルシステムを利用して、その共有対象のファイル（又はデータ）を磁気ディスク装置上に配置して処理することが必要である。この場合には、プロセスが、ファイル（又はデータ）に対する処理を入出力装置を対象とする I / O 処理のコマンドを使用してプログラミングされている必要がある。このようにファイル装置（磁気ディスク装置）に記憶されているファイル（又はデータ）のデータ処理する場合は、メモリ上に配置されたファイル（又はデータ）を処理する場合に比べて、プログラムの記述が複雑になる。これは、メモリ上に配置されたデータに対する処理であれば、load 命令や store 命令を使用して簡単なプログラム記述で処理が記述できる。

【 0 0 1 1 】

しかし、これに対して磁気ディスク装置上に配置されたファイルやデータを処理する場合には、I / O 処理をするための複雑なコマンドを使用したプログラムを記述する必要があり、並列プログラムにて記述するのが困難である。また、メモリ上に配置されたデータに対するデータ処理に比べて、磁気ディスク装置などのファイル装置（I / O 装置）上に配置されたデータを処理する場合には、その処理時間が多くかかり、処理速度が落ちるという問題もある。

【 0 0 1 2 】

以上の説明したように、クラスタ共有ファイルシステムを実装した複数のサーバーコンピュータが疎結合されたクラスタシステムでは、磁気ディスク装置などのファイル装置上に記憶されたファイルへの共有アクセスは可能であるが、主メモリへの共有アクセスは可能でないため、SMP 型並列計算機と比べると、並列プログラムにて記述するのが困難であるという課題がある。

【0013】

本発明は、クラスタ共有ファイルシステムを実装した複数のサーバーコンピュータが疎結合されたクラスタシステムにおいて、並列プログラムの記述が容易なクラスタシステムを提供することを目的とする。

【0014】

【課題を解決するための手段】

本発明は、複数の電子計算機と、共有ディスク装置と、前記複数の電子計算機上で動作するプロセスから前記共有ディスク装置に記録されたファイルに対して共有アクセスするためにロック機能によりデータの一貫性を保持するための排他制御をするクラスタ共有ファイルシステムとを持ったクラスタシステムであって、前記クラスタ共有ファイルシステムが管理するファイルを前記プロセスのアドレス空間内にマッピングして共有メモリを設定するためのクラスタ共有メモリ設定手段と、前記クラスタ共有ファイルシステムのロック機能を前記共有メモリにアロケーションして前記共有メモリ上でデータの一貫性を保持するための排他制御を可能とするクラスタ共有メモリ用ロックアロケーション手段とを備えたことを特徴とする。

【0015】

本発明によれば、クラスタ共有ファイルシステムを実装した複数のサーバーコンピュータが疎結合されたクラスタシステムにおいて、並列プログラムの記述を容易にすることができる。

【0016】

【発明の実施の形態】

本発明のポイントは、クラスタシステムにおいて、アプリケーションプログラムを実行することで生成されるプロセスが共有ディスク装置に記録されたファイルを、クラスタ共有ファイルシステムを用いてプロセスが主メモリ上に配置されているアドレス空間内に仮想的に設けたクラスタ共有メモリ（分散共有メモリ）領域上にマッピングすることで、ファイルへのアクセスを主メモリへのアクセスとして処理することである。

【0017】

以下、図面を用いて、本発明の一実施形態を詳細に説明する。図1は、本発明のクラスタシステムを示す図である。図1において、クラスタシステム10は、ファイバ・チャンネルで構成されているストレージエリア・ネットワーク（SAN）15にサーバーコンピュータ11、サーバーコンピュータ12、サーバーコンピュータ13、サーバーコンピュータ14と磁気ディスク装置で構成される共有ディスク装置16が接続されて構成されている。

【0018】

各サーバーコンピュータ11～14は、図2に示すように構成されている。図2では、サーバーコンピュータ11～14の代表として、サーバーコンピュータ11の構成を説明する。図2において、サーバーコンピュータ11には、主メモリ111、クラスタ共有メモリ用ロック/クラスタ共有ファイルシステム用ロック変換テーブル113、クラスタ共有メモリ用ロック操作手段114、クラスタ共有メモリ用ページ操作手段115、クラスタ共有ファイルシステム用ロック機構116、クラスタ共有ファイルシステム117、クラスタ共有メモリ/クラスタ共有ファイル変換テーブル118、更新ページリスト119、クラスタ共有メモリマップ手段120、クラスタ共有メモリアンマップ手段121、クラスタ共有メモリ用ロックアロケーション手段122、クラスタ共有メモリ用ロック解放手段123が設けられて構成されている。サーバーコンピュータ11では、あるアプリケーションプログラムが実行されることで生成されるプロセス112が主メモリ111上のアドレス空間内に配置されて図示しないプロセッサにより実行されて動作している。

【0019】

図3は、プロセス112の詳細を示す図である。図3において参照符号Aは、プロセス112のアドレス空間内におけるアドレスを示す。プロセス112には、アドレスAを先頭として複数のページP0、P1、P2、P3、P4、P5が設けられてる。このプロセス112内に仮想的にクラスタ共有メモリが設けられる。

【0020】

クラスタ共有メモリ用ロック/クラスタ共有ファイルシステム用ロック変換テ

ーブル113は、図4に示すようにクラスタ共有メモリ用ロックID欄113aとクラスタ共有ファイルシステム用ロックID欄113bとから構成されている。このクラスタ共有メモリ用ロック/クラスタ共有ファイルシステム用ロック変換テーブル113にクラスタ共有メモリ用ロックIDとクラスタ共有ファイルシステム用ロックIDとを登録することにより、両者の対応を関係づける。

【0021】

クラスタ共有メモリ用ロック操作手段114は、クラスタ共有メモリ内に設けられる各ページに記憶されたデータの一貫性を確保するための排他制御をするためにロックを設定（ロックする）し、また設定したロックを解除（アンロック）するための操作手段である。クラスタ共有メモリ用ページ操作手段115は、クラスタ共有メモリに設けられ各ページに対するデータの記憶、各ページに記憶されたデータの読み出しを行う操作手段である。

【0022】

クラスタ共有ファイルシステム117は、共有ディスク装置16に記録されたファイルをサーバコンピュータ11、サーバコンピュータ12、サーバコンピュータ13、サーバコンピュータ14とで共有するためのファイルシステムである。クラスタ共有ファイルシステム用ロック機構116は、共有ディスク装置16に記録されたファイルをサーバコンピュータ11、サーバコンピュータ12、サーバコンピュータ13、サーバコンピュータ14とで共有する際に、データの一貫性を確保するための排他制御をするためにロックを設定（ロックする）し、また設定したロックを解除（アンロック）するための操作を行う機構である。

【0023】

クラスタ共有メモリ/クラスタ共有ファイル変換テーブル118は、図5に示すように、アドレス欄118a、サイズ欄118b、ファイル名欄118c、ファイル記述子118d、オフセット欄118eとから構成されている。このクラスタ共有メモリ/クラスタ共有ファイル変換テーブル118は、共有ディスク装置16に記録されたファイルをクラスタ共有メモリに対応付け（マッピング）するために設けられている。クラスタ共有メモリマップ手段120は、プロセス112のアドレス空間内にクラスタ共有メモリを仮想的に設けるための手段である

【0024】

クラスタ共有メモリアンマップ手段121は、プロセス112のアドレス空間内に仮想的に設けられたクラスタ共有メモリを解放するための手段である。

クラスタ共有メモリ用ロックアロケーション手段122は、クラスタ共有メモリ用ロックと一対一に対応するクラスタ共有ファイルシステム用ロックとをアロケーションするための手段である。クラスタ共有メモリ用ロック解放手段123は、クラスタ共有メモリ用ロックアロケーション手段122により、アロケーションされたクラスタ共有メモリ用ロックとクラスタ共有ファイルシステム用ロックとを解放するための手段である。

【0025】

図6は、更新ページリスト119の詳細を示す図である。この更新ページリスト119は、共有メモリの中のデータが更新されたページのページ番号を記録するためのリストである。

【0026】

以下、図7を用いてサーバーコンピュータ11で実行されているプロセス112が共有ディスク装置16に記録されているファイルに対してアクセスする場合の動作を詳細に説明する。

【0027】

図7は、クラスタ共有メモリを用いたクラスタ共有ファイルへのアクセス動作の処理手順を示すフローチャート図である。まず、プロセス112が共有ディスク装置16に記録されている共有ファイルに対してアクセスしようとする場合には、プロセス112のアドレス空間内に仮想的に共有メモリを設け、この共有メモリ上にアクセス対象の共有ファイルをマッピングする（ステップS70）。具体的には、プロセス112がクラスタ共有メモリマップ手段120にクラスタ共有メモリマップ操作を指示する。このプロセス112からの指示に基づいてクラスタ共有メモリマップ手段120は、プロセス120が配置された主メモリ111上のアドレス空間内に共有メモリをマッピングする。

【0028】

このクラスタ共有メモリマップ手段120による共有メモリのマッピング操作手順を図8に示すフローチャートを用いて説明する。まず、ステップS80で、プロセス112のアドレス空間内にクラスタ共有メモリのための領域をアロケーションし、その領域内の全てのページをアクセス不許可に設定する。アロケーションは、プロセス112にC言語の関数 `m a l l o c ()` を記述することで実施できる。

【0029】

次に、ステップS81で、アクセス対象の共有ファイルのプロセス112から指定されたオフセット位置から指定されたサイズ分のデータをステップS80でアロケーションした領域にマッピングしたことをクラスタ共有メモリ/クラスタ共有ファイル変換テーブル118に登録する。この登録の結果として、図5に示すようにプロセス112のアドレスAからサイズLの範囲にファイル名が「DDDD」で、ファイル記述子が「7」で、オフセット「0」に登録する。ここまでの処理で、プロセス112が共有ファイルへのアクセスを共有メモリへのアクセスとして処理するための環境を整えたことになる。

【0030】

続いてステップS71において、プロセス112が共有ファイルへのアクセスを共有メモリへのアクセスとして処理するのに必要な排他制御処理をするためにロックを取得する。そこでロック取得の前準備として、プロセス112は、クラスタ共有メモリ用ロックとクラスタ共有ファイル用ロックとを対応づけるために、クラスタ共有メモリ用ロックアロケーション手段122にクラスタ共有メモリ用ロックアロケーション操作を指示する。

【0031】

図10にクラスタ共有メモリ用ロックアロケーション手段122によるクラスタ共有メモリ用ロックアロケーション操作手順のフローチャートを示す。ステップS100において、クラスタ共有メモリ用ロックと一対一に対応するクラスタ共有ファイルシステム用ロックをアロケーションするために、図4に示すクラスタ共有メモリ用ロック/クラスタ共有ファイルシステム用ロック変換テーブル113にクラスタ共有メモリ用ロックのID番号とクラスタ共有ファイルシステム

用ロックのID番号とを対応づけて登録する。

【0032】

次にプロセス112は、ロックを取得するために、クラスタ共有メモリ用ロック操作手段114にロックの取得を指示する。このロック取得の指示を受けたクラスタ共有メモリ用ロック操作手段114のロック操作の処理手順を図12に示したフローチャートを用いて説明する。

【0033】

まずステップS120において、クラスタ共有メモリ上の全てのページをアクセス不許可に設定する。次にステップS121において、図4に示したクラスタ共有メモリ用ロック/クラスタファイルシステム用ロック変換テーブル113を用いて、ロック操作されるクラスタ共有メモリ用ロックをクラスタ共有ファイルシステム用ロックに変換する。続いて、ステップS122において、クラスタ共有メモリ用ロック操作手段114は、クラスタ共有ファイルシステム用ロック機構116に指示して、変換したクラスタファイルシステム用ロックをロック操作してロックを取得する。

【0034】

ステップS71でロック取得が成功した場合には、ステップS72に進む。もし、ロック取得が失敗した場合には、換言すると既に他のプロセスが同様な処理により、実体としての共有ファイルのロックを取得している場合には、ロックが取得できないので、ロックが解放されて、取得できるまで処理を待機しステップS71の処理を再び続ける

【0035】

続いてステップS72において、プロセス112は、クラスタ共有メモリ用ページ操作手段115に指示して、共有メモリへのREADアクセス（共有メモリに対するload命令の実行）又はWRITEアクセス（共有メモリに対するstore命令の実行）を実行させる。このクラスタ共有メモリ用ページ操作手段115が共有メモリへアクセスを実行すると、この共有メモリの全てのページはアクセス不許可に設定されているため、ページフォールトが発生する。クラスタ共有メモリ用ページ操作手段115がREADアクセスした場合には、READ

ページフォールトが発生する。また、クラスタ共有メモリ用ページ操作手段115がWRITEアクセスした場合には、WRITEページフォールトが発生する。

【0036】

READページフォールトが発生した場合のクラスタ共有メモリ用ページ操作手段115の処理手順を図14に示す。WRITEページフォールトが発生した場合のクラスタ共有メモリ用ページ操作手段115の処理手順を図15に示す。

【0037】

図14を参照して、READページフォールトが発生した場合のクラスタ共有メモリ用ページ操作手段115の処理手順を説明する。まず、ステップS140で、ページフォールトが発生した共有メモリ内のページのデータをクラスタ共有ファイルシステム117によりクラスタ共有ファイル（共有ディスク装置16）の対応する部分から当該共有メモリ内のページに読み込む。この時、ページのデータをクラスタ共有ファイルのどの部分から読み込むかは、クラスタ共有メモリ/クラスタ共有ファイル変換テーブル118に登録されたデータに基づいて求める。続いて、ステップS141で、ページフォールトが発生したページをREAD可能に設定する。

【0038】

次に図15を参照して、WRITEページフォールトが発生した場合のクラスタ共有メモリ用ページ操作手段115の処理手順を説明する。まず、ステップS150において、ページフォールトが発生したページがREAD可能に設定されているかどうかを調べる。READ可能な場合には、ステップS152へ進む。READ可能でない場合には、ステップS151に進む。

【0039】

ステップS151では、ページフォールトが発生した共有メモリ内のページのデータをクラスタ共有ファイルシステム117によりクラスタ共有ファイル（共有ディスク装置16）の対応する部分から当該共有メモリ内のページに読み込む。この時、ページのデータをクラスタ共有ファイルのどの部分から読み込むかは

、クラスタ共有メモリ/クラスタ共有ファイル変換テーブル118に登録されたデータに基づいて求める。

【0040】

続いて、ステップS152で、ページフォールトが発生した共有メモリ内のページをREAD/WRITE可能に設定する。最後に、ステップS153で、ページフォールトが発生した共有メモリ内のページの番号を図6に示す更新ページリスト119に登録する。クラスタ共有メモリ用ページ操作手段115は、このようにページフォールト処理をした後、実際にクラスタ共有メモリへアクセスする。

【0041】

このようにステップS72で共有メモリへのアクセスをした後、続くステップS73において、プロセス112は、クラスタ共有メモリ用ロックを解放するためにクラスタ共有メモリ用ロック操作手段114に指示して、クラスタ共有メモリロックのアンロック操作を実行させる。

【0042】

ここで図13を参照して、クラスタ共有メモリ用ロック操作手段114によるクラスタ共有メモリロックのアンロック操作手順を説明する。図13は、クラスタ共有メモリ用ロック操作手段114によるクラスタ共有メモリロックのアンロック操作の操作手順を示すフローチャート図である。

【0043】

まず、ステップS130において、図6に示した更新ページリスト119に登録されている全ての更新ページのデータをクラスタ共有ファイルシステム117により共有ディスク装置16に記録されているクラスタ共有ファイルの該当部分に書き込む。この時、更新ページをクラスタ共有ファイルのどの部分に書き込むかは、クラスタ共有メモリ/クラスタ共有ファイル変換テーブル118を用いて求める。続いて、ステップS131において、更新ページリスト119をクリアする。

【0044】

以上で図7に示したクラスタ共有メモリを用いたクラスタ共有ファイルへのア

クセス動作の処理手順の説明を終了する。ここで、プロセス112は、これまでの説明でアクセスした共有ファイルをまだアクセスする場合には、図4に示したクラスタ共有メモリ用ロック／クラスタ共有ファイル用ロック変換テーブル113のエントリを削除せずに残しておく。このエントリを残しておけば、次にロックを取得する場合には、クラスタ共有メモリ用ロック操作手段114がクラスタ共有ファイルシステム用ロック機構116にファイルのロック取得を依頼するだけでロック取得ができる。

【0045】

もし、プロセス112がこれまでの説明でアクセスした共有ファイルを以後アクセスすることがない場合には、プロセス112は、クラスタ共有メモリ用ロック解放手段123に指示してクラスタ共有メモリ用ロックを解放させる。図11は、クラスタ共有メモリ用ロック解放手段123によるクラスタ共有メモリ用ロックを解放させる処理手順を示すフローチャート図である。図11において、ステップS110で、プロセス112が指定したクラスタ共有メモリ用ロックに関するエントリを図4に示すクラスタ共有メモリ用ロック／クラスタ共有ファイルシステム用ロック変換テーブル113から抹消する。これにより、クラスタ共有メモリ用ロックが解放される。

【0046】

最後にプロセス112が共有メモリへのアクセスとして処理すべき全ての共有ファイルへのアクセスが終了した場合には、プロセス112がクラスタ共有メモリアンマップ手段121に指示してプロセスのアドレス空間内に仮想的に設定した共有メモリのマッピングを解除させる。すなわち、クラスタ共有メモリをアンマップさせる。

【0047】

図9は、クラスタ共有メモリアンマップ手段121がクラスタ共有メモリをアンマップする処理手順を示すフローチャート図である。図9において、まず、ステップS90において、プロセス112のアドレス空間内にアロケーションしたクラスタ共有メモリのための領域を解放する。続いて、ステップS91において、アンマップする領域に関するエントリを図5に示すクラスタ共有メモリ／クラ

スタ共有ファイル変換テーブル118から抹消する。

【0048】

次に図16を用いて本発明において、2つのサーバーコンピュータが共有ディスク装置に記録されている同じファイルに対するアクセスをする場合の動作を説明する。図16は、サーバーコンピュータAとサーバーコンピュータBと共有ディスク装置Cとから構成されるクラスタシステムの動作を説明するためのシステム構成を示す図である。。

【0049】

図16において、サーバーコンピュータAとサーバーコンピュータBとは、それぞれ共有ディスク装置Cに接続されている。サーバーコンピュータAでは、プロセスAが動作している。また、サーバーコンピュータBでは、プロセスBが動作している。これらのプロセスAとプロセスBは、それぞれ図示を省略したクラスタ共有ファイルシステムで管理されている共有ディスク装置Cに記録されたクラスタ共有ファイルDを先に説明したように自身のアドレス空間内にマッピングしているものとする。クラスタ共有ファイルDの中にはデータ領域Xが存在する。

【0050】

このように設定された状況の下に、プロセスAが(1)～(3)の処理を実行し、次にプロセスBが(4)～(6)の処理を実行し、更に再びプロセスA(7)～(9)の処理を実行するとする。まずプロセスAは(1)でクラスタ共有メモリ用ロックを取得する。これによりクラスタ共有ファイルDがクラスタ共有メモリとしてマップされた領域のページは全てアクセス不許可になる。次に(2)でクラスタ共有メモリ上のデータ領域Xに数値「1」を加えようとする。するとWRITEページフォールトが発生する。クラスタ共有メモリ操作手段は、この領域がREAD不可能なので、このデータ領域を含むページの内容をクラスタ共有ファイルDから読みこむ。

【0051】

そして、そのページをREAD/WRITE可能に設定し、ページフォールト処理を終了する。ページフォールトから戻った後、プロセスAはデータ領域Xの

値に1を加える。そして(3)でクラスタ共有メモリ用ロックを解放する。

これにより、クラスタ共有メモリ用ロック操作手段は、更新ページであるデータ領域Xを含むページの内容をクラスタ共有ファイルDに書き戻す。データ領域Xの初期値が「0」だったとすると、この時点でデータ領域Xの値は「1」となる。

【0052】

次にプロセスBは(4)でクラスタ共有メモリ用ロックを取得する。

これによりクラスタ共有ファイルD4がクラスタ共有メモリとしてマップされた領域のページは全てアクセス不許可になる。次に(5)でクラスタ共有メモリ上のデータ領域Xに数値「1」を加えようとする。するとWRITEページフォールトが発生する。クラスタ共有メモリ操作手段は、この領域がREADが不可能なので、このデータ領域を含むページの内容をクラスタ共有ファイルDから読みこむ。そして、そのページをREAD/WRITE可能に設定し、ページフォールト処理を終了する。ページフォールトから戻った後、プロセスBはデータ領域Xの値に数値「1」を加える。そして(6)でクラスタ共有メモリ用ロックを解放する。これにより、クラスタ共有メモリ用ロック操作手段は、更新ページであるデータ領域Xを含むページの内容をクラスタ共有ファイルDに書き戻す。この時点でデータ領域Xの値は「2」になる。

【0053】

最後にプロセスAは(7)でクラスタ共有メモリ用ロックを取得する。これによりクラスタ共有ファイルDがクラスタ共有メモリとしてマップされた領域のページは全てアクセス不許可になる。次に(8)でクラスタ共有メモリ上のデータ領域Xに1を加えようとする。するとWRITEページフォールトが発生する。クラスタ共有メモリ操作手段は、この領域がREAD/WRITE不可能なので、このデータ領域を含むページの内容をクラスタ共有ファイルDから読みこむ。そして、そのページをすREAD/WRITE可能に設定し、ページフォールト処理を終了する。ページフォールトから戻った後、プロセスAはデータ領域Xの値に数値「1」を加える。そして(9)でクラスタ共有メモリ用ロックを解放する。これにより、クラスタ共有メモリ用ロック操作手段は、更新ページであるデ

ータ領域Xを含むページの内容をクラスタ共有ファイルDに書き戻す。この時点でデータ領域Xの値は3になる。

【0054】

【発明の効果】

本発明を適用することにより、クラスタ共有ファイルシステムを持つクラスタシステムで、ファイルへの共有アクセスと共に、メモリへの共有アクセスも可能になる。

更に本発明では、クラスタ共有ファイルシステムを用いることで、クラスタ共有メモリを安価に実現することができる。また、クラスタ共有ファイルをクラスタ共有メモリとしてマッピングするので、更新データに永続性が持たれる。更に、本発明では、クラスタ共有メモリを、クラスタ共有ファイルシステムを用いて実現するため、クラスタ共有ファイルを分散共有メモリとしてマッピングし、メモリとしてのアクセス（load命令/store命令）とファイルとしてのアクセス（readシステムコール/writeシステムコール）を並列に実行することができるようになる。

【図面の簡単な説明】

【図1】

本発明のクラスタシステム10を示す図である。

【図2】

サーバーコンピュータ11の構成を示す図である。

【図3】

プロセス112の詳細を示した図である。

【図4】

クラスタ共有メモリ用ロック/クラスタ共有ファイルシステム用ロック変換テーブル113の詳細を示した図である。

【図5】

クラスタ共有メモリ/クラスタ共有ファイル変換テーブル118の詳細を示した図である。

【図6】

更新ページリスト 119 の詳細を示した図である。

【図 7】

クラスタ共有メモリを用いたクラスタ共有ファイルへのアクセス動作の処理手順を示すフローチャート図である。

【図 8】

クラスタ共有メモリマップ手段 120 による共有メモリのマッピング操作手順を示すフローチャート図である。

【図 9】

クラスタ共有メモリアンマップ手段 121 がクラスタ共有メモリアンマップする処理手順を示すフローチャート図である。

【図 10】

クラスタ共有メモリ用ロックアロケーション手段 122 によるクラスタ共有メモリ用ロックアロケーション操作手順を示すフローチャート図である。

【図 11】

クラスタ共有メモリ用ロック解放手段 123 によるクラスタ共有メモリ用ロックを解放させる処理手順を示すフローチャート図である。

【図 12】

クラスタ共有メモリ用ロック操作手段 114 のロック操作の処理手順を示すフローチャート図である。

【図 13】

クラスタ共有メモリ用ロック操作手段 114 によるクラスタ共有メモリロックのアンロック操作の操作手順を示すフローチャート図である。

【図 14】

READ ページフォールトが発生が発生した場合のクラスタ共有メモリ用ページ操作手段 115 の処理手順を示すフローチャート図である。

【図 15】

WRITE ページフォールトが発生が発生した場合のクラスタ共有メモリ用ページ操作手段 115 の処理手順を示すフローチャート図である。

【図 16】

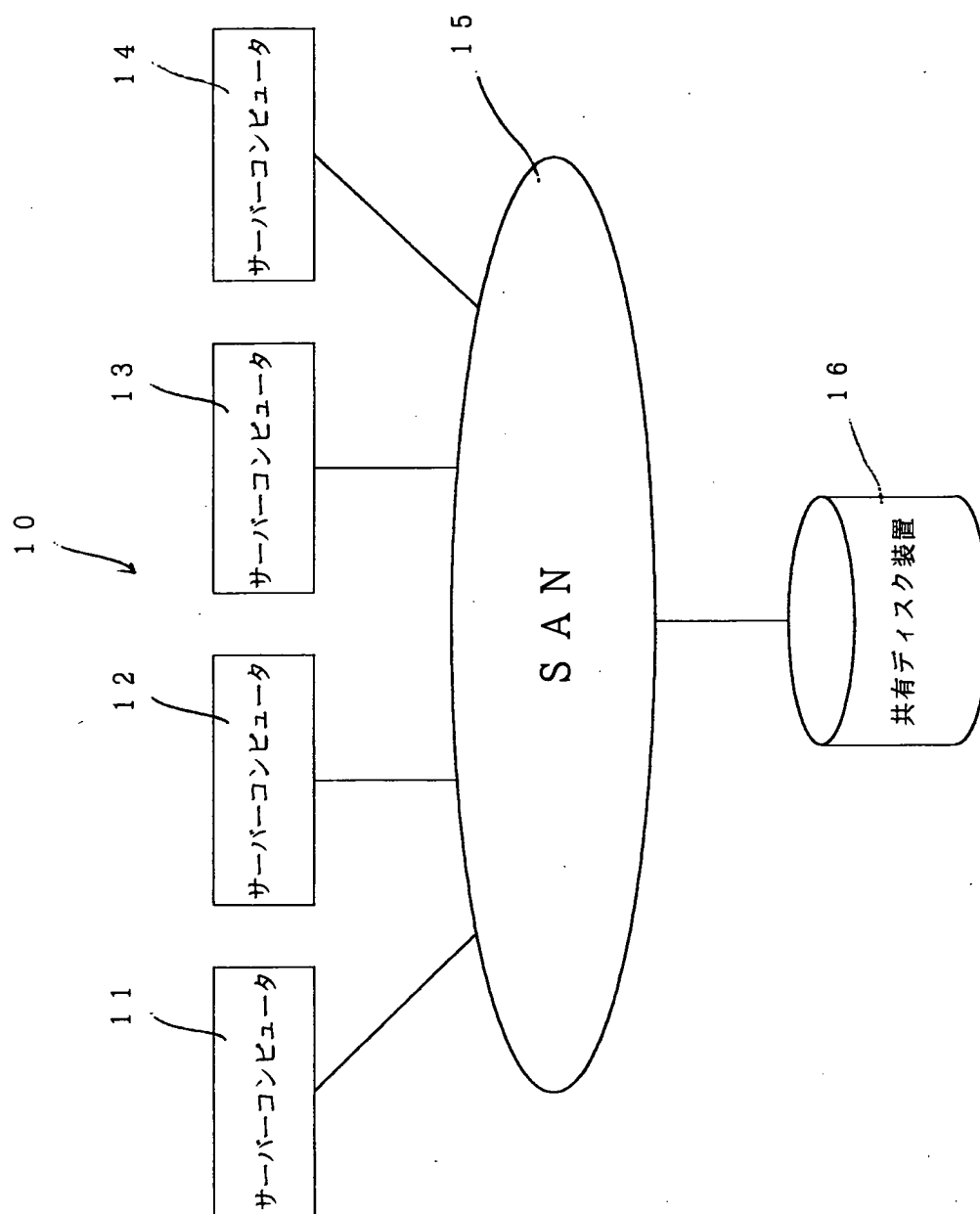
2つのサーバーコンピュータが共有ディスク装置に記録されている同じファイルに対するアクセスをする場合の動作を説明するためのシステム構成を示す図である。

【符号の説明】

- 1 1 ……サーバーコンピュータ
- 1 2 ……サーバーコンピュータ
- 1 3 ……サーバーコンピュータ
- 1 4 ……サーバーコンピュータ
- 1 5 ……S A N
- 1 6 ……共有ディスク装置

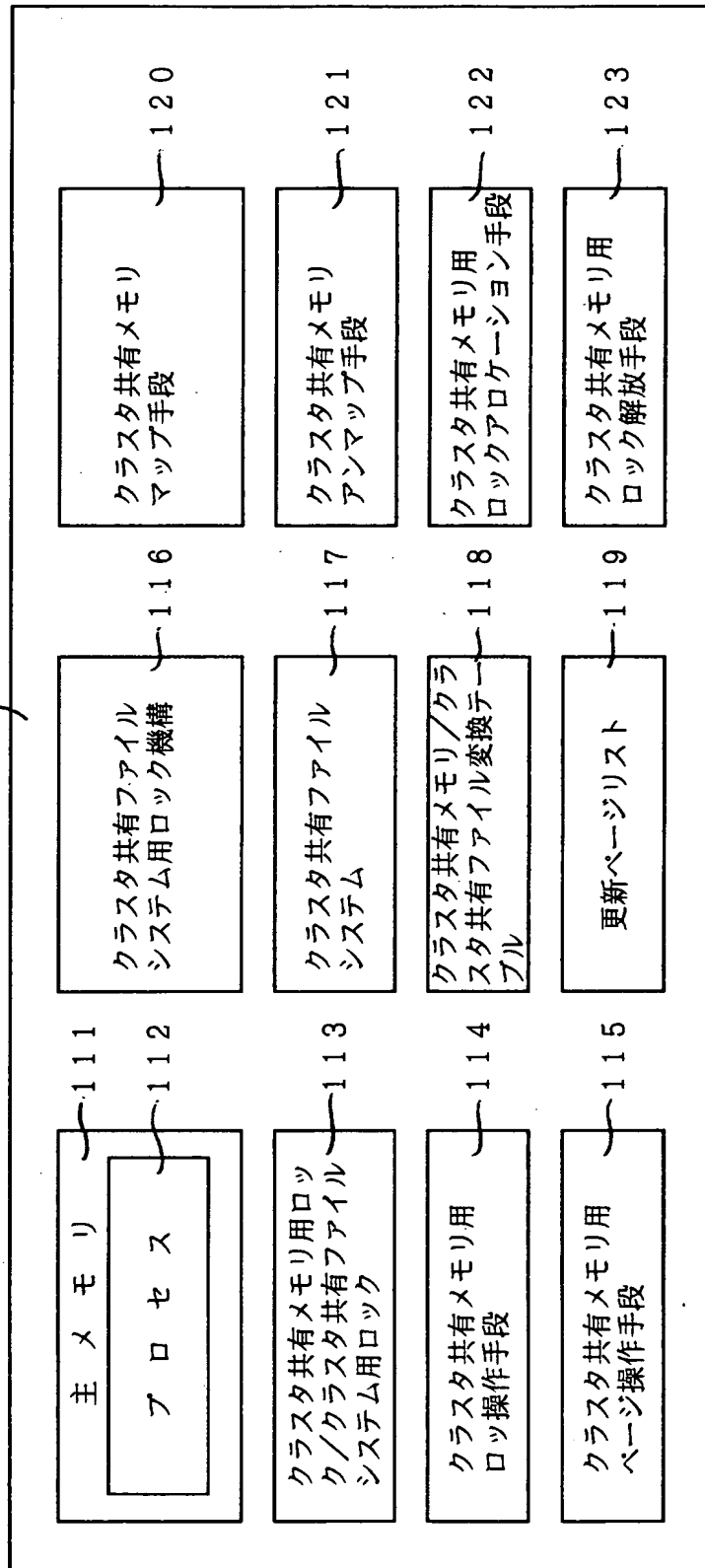
【書類名】 図面

【図1】

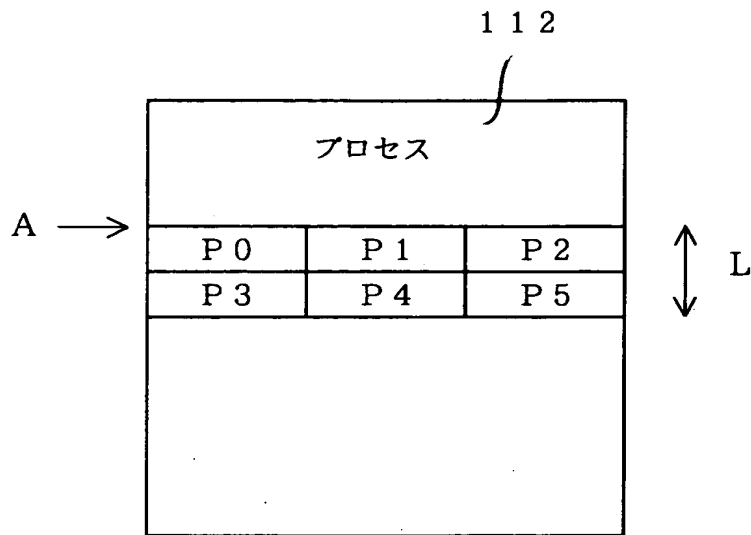


【図2】

11



【図 3】



【図 4】

113	↓	クラスタ共有メモリ用ロックID	クラスタ共有ファイルシステム用ロックID
113a	↓	18	1018

【図 5】

118

↓

118a 118b 118c 118d 118e

アドレス	サイズ	ファイル名	ファイル記述子	オフセット
A	L	DDDD	7	0

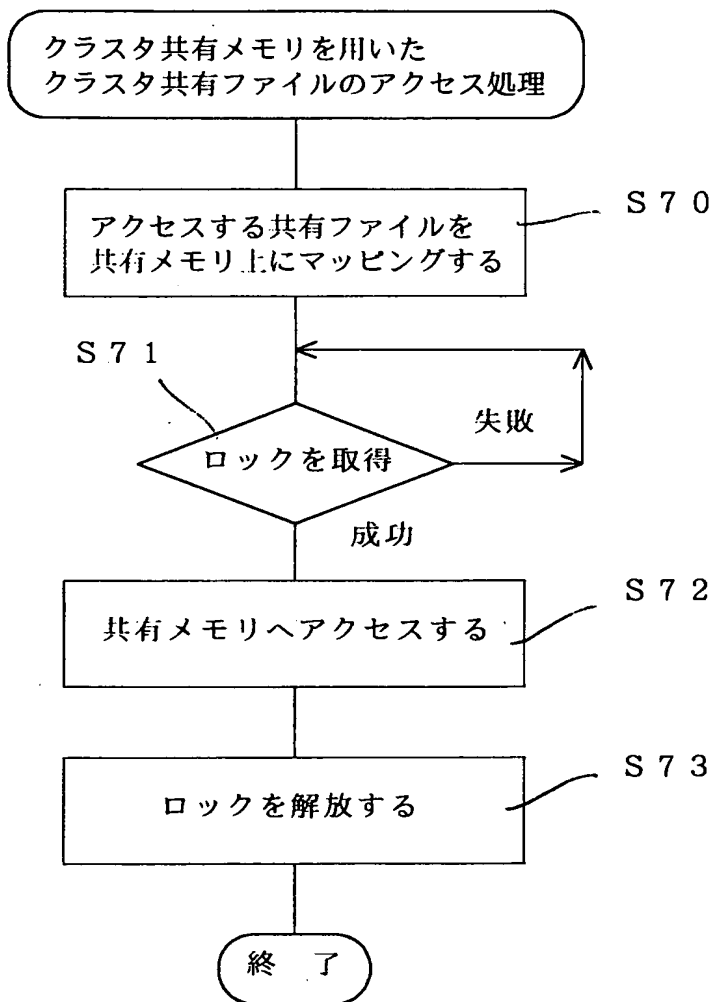
【図 6】

119

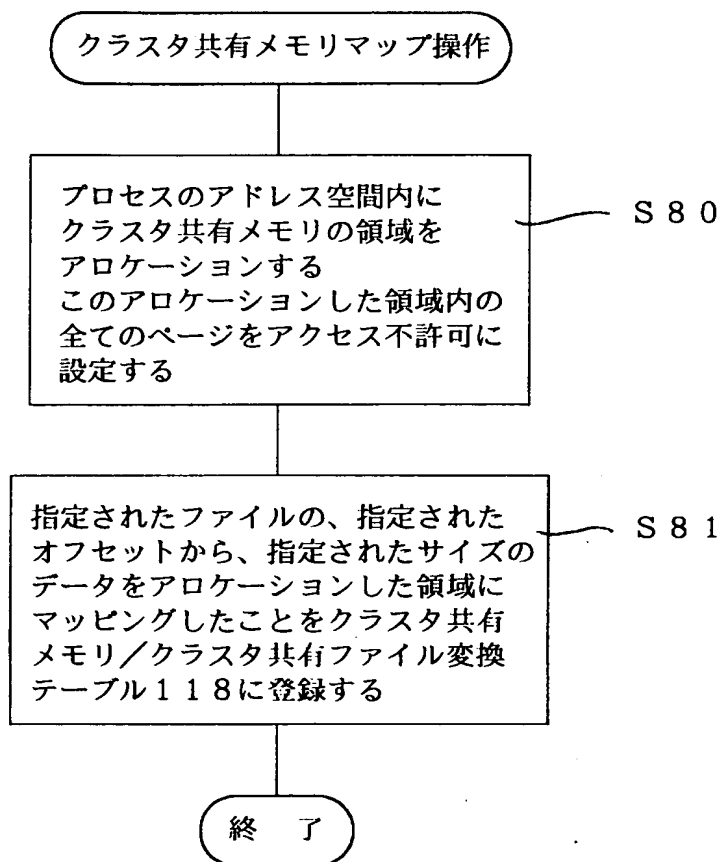
↓

P4	P2		

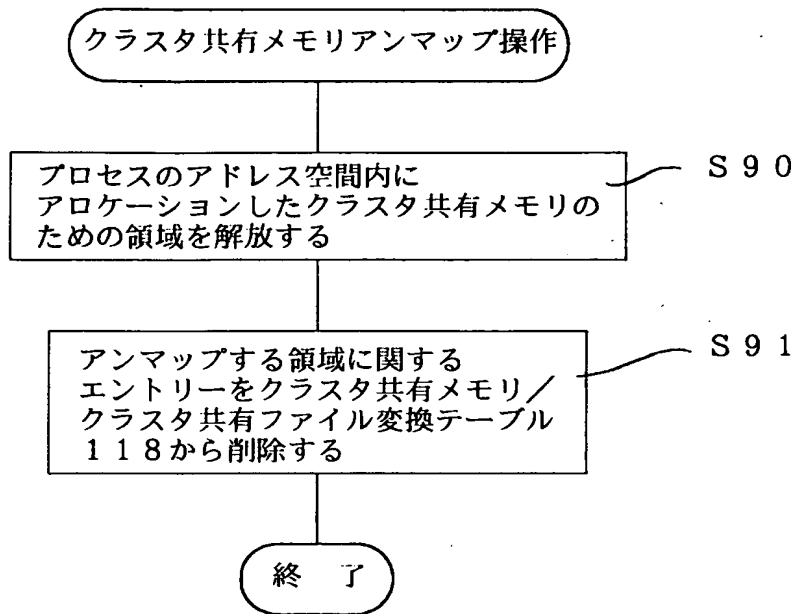
【図 7】



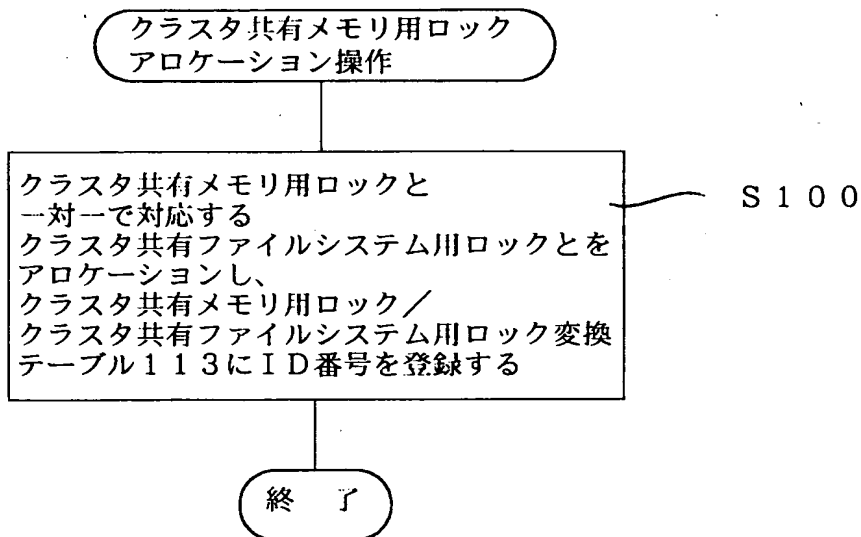
【図 8】



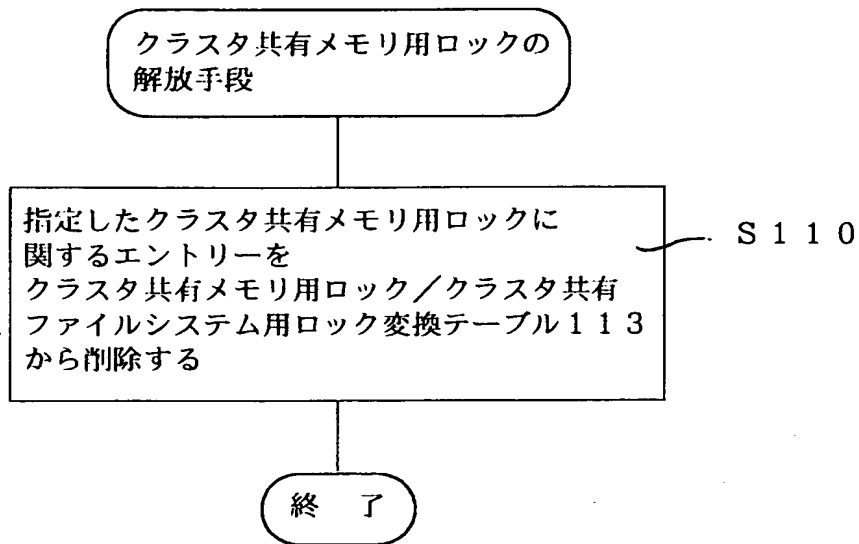
【図 9】



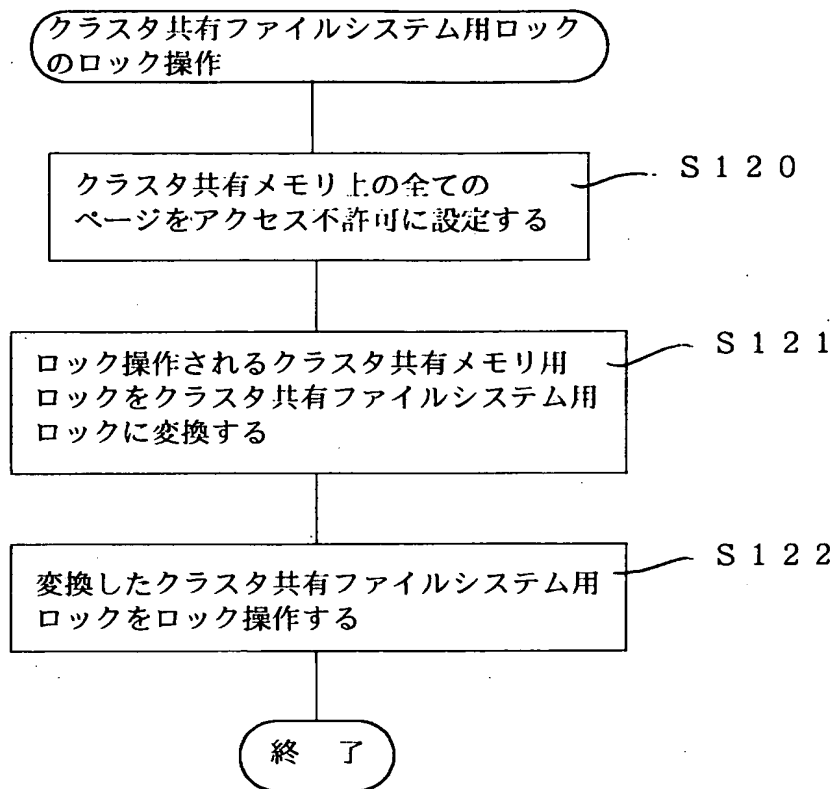
【図 1 0】



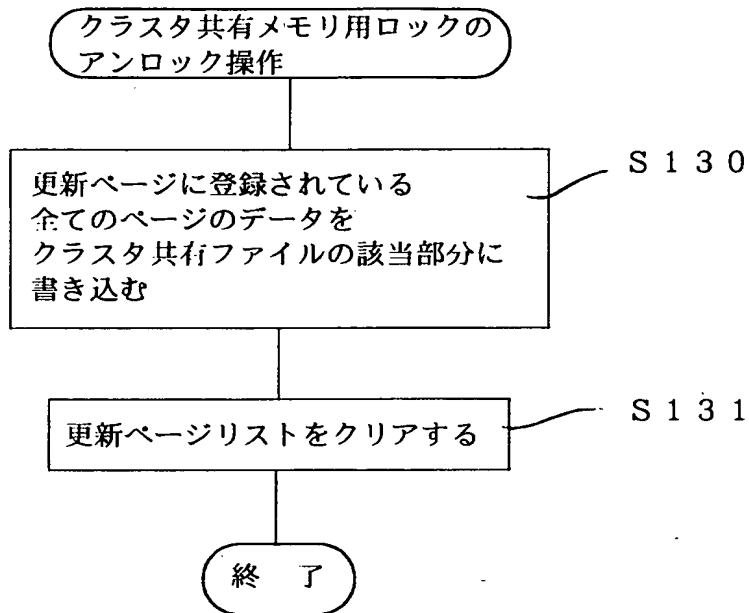
【図 1 1】



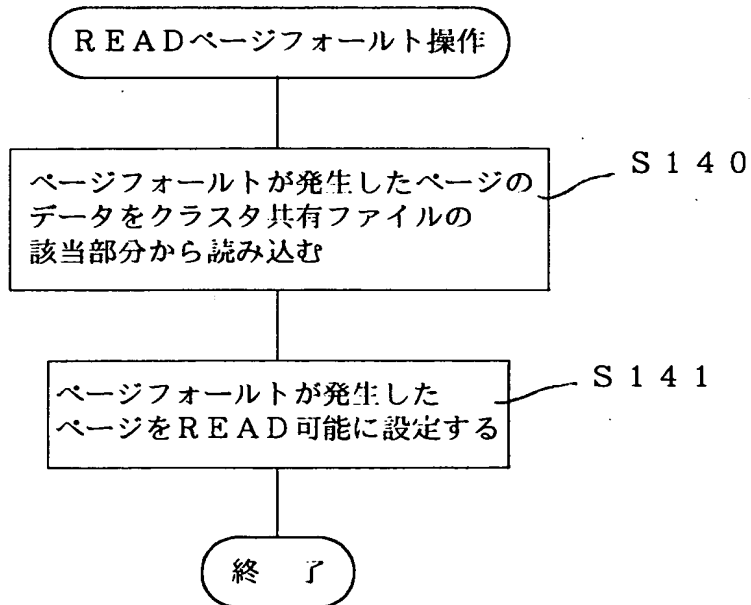
【図 1 2】



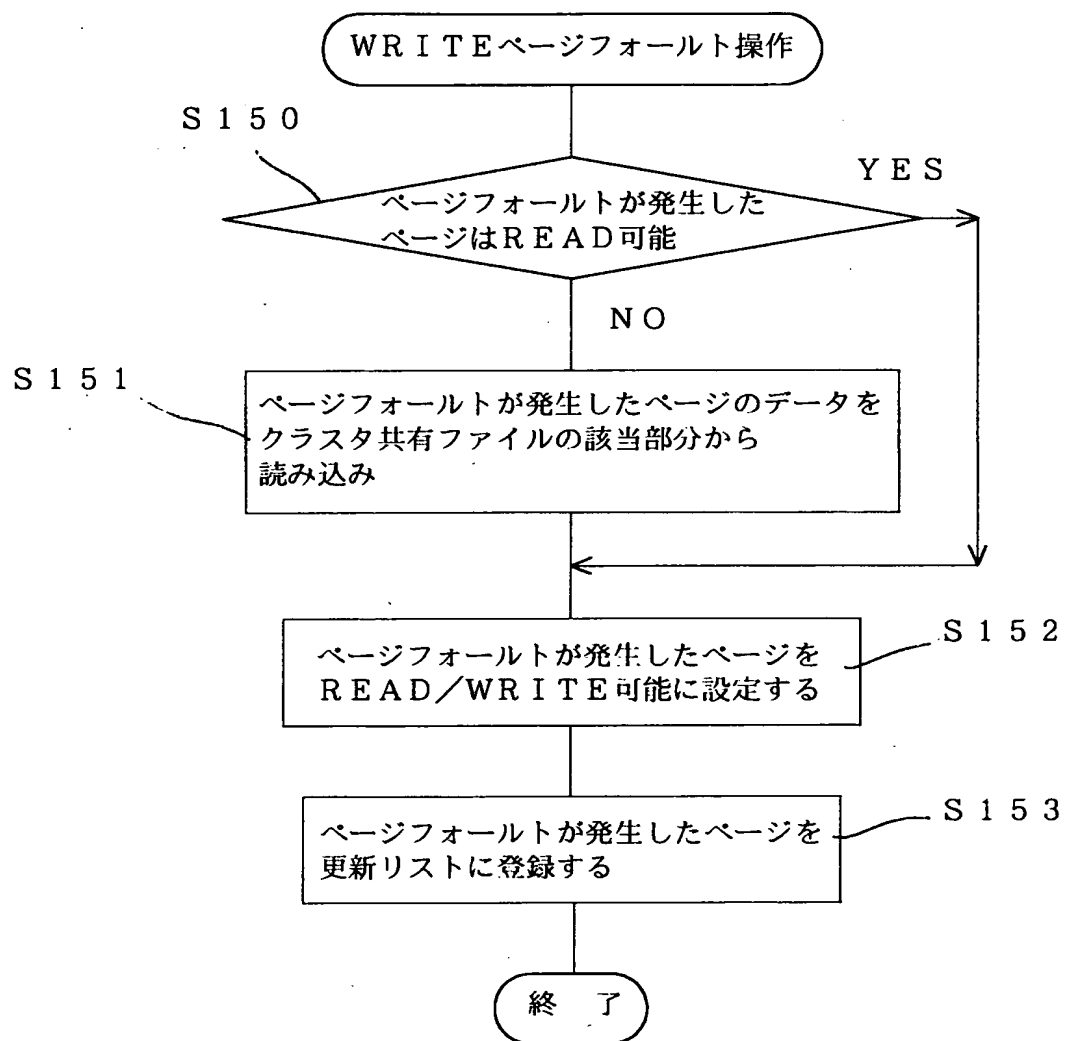
【図 1 3】



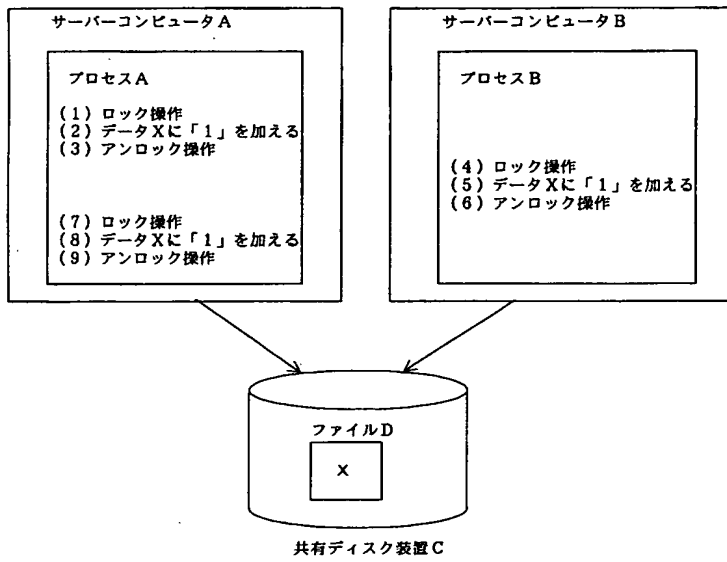
【図 1 4】



【図15】



【図 16】



【書類名】 要約書

【要約】

【課題】 本発明は、クラスタ共有ファイルシステムを実装した複数のサーバーコンピュータが疎結合されたクラスタシステムにおいて、並列プログラムの記述が容易なクラスタシステムを提供することを目的とする。

【解決手段】 クラスタシステムにおいて、アプリケーションプログラムを実行することで生成されるプロセスが共有ディスク装置に記録されたファイルをクラスタ共有ファイルシステムを用いてプロセスが主メモリ上に配置されているアドレス空間内に仮想的に設けたクラスタ共有メモリ（分散共有メモリ）領域上にマッピングすることで、ファイルへのアクセスを主メモリへのアクセスとして処理する。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願2001-071680
受付番号	50100360701
書類名	特許願
担当官	第七担当上席 0096
作成日	平成13年 3月15日

<認定情報・付加情報>

【提出日】	平成13年 3月14日
-------	-------------

出 願 人 履 歴 情 報

識別番号 [000003078]

1. 変更年月日 1990年 8月22日
[変更理由] 新規登録
住 所 神奈川県川崎市幸区堀川町72番地
氏 名 株式会社東芝
2. 変更年月日 2001年 7月 2日
[変更理由] 住所変更
住 所 東京都港区芝浦一丁目1番1号
氏 名 株式会社東芝